



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/761,139	01/20/2004	Thomas R. Gumz	260-134	8821
44185	7590	12/05/2007		
LOTUS AND RATIONAL SOFTWARE			EXAMINER	
McGuinness & Manaras LLP			BLACKWELL, JAMES H	
125 NAGOG PARK			ART UNIT	PAPER NUMBER
ACTON, MA 01720			2176	
			MAIL DATE	DELIVERY MODE
			12/05/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/761,139

Applicant(s)

GUMZ ET AL.

Examiner

James H. Blackwell

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 27 September 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,2,4-8,10-12 and 17-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1,2,4-8,10-12 and 17-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 24 June 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application
- ☐ Other: _____

DETAILED ACTION

1. This Office Action is in response to an amendment filed 03/12/2007 with a filing date of **01/20/2004**.
2. Claims 1-2, 4-8, 10-12, and 17-20 remain pending.
3. Claims 9, and 14-16 have been cancelled with this amendment.
4. Claims 1, 6, 12, and 17 are independent claims.
5. Objections to Claims 6, 14, and 19 have been withdrawn. Claim 14 was cancelled; Claims 6, and 19 were amended to overcome their objections.
6. Rejection of Claim 14 under 35 USC § 101 has been withdrawn. Claim 14 was cancelled.

Claim Objections

Claim 1 is objected to because of the following informalities:

- The phrase "*parsing said document to determine if certain of said plurality is said custom tag*" in Line 5 should be amended to — parsing said document to determine if certain of said plurality of tags are *[[is]]*said custom tag — because that is how the element is previously identified (see Line 3, "*plurality of tags*") and so that the claim reads in grammatically-correct English.

Appropriate correction is required.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-2, 4-8, 10-12, and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lennox (Gareth Lennox, "Client Side Includes", copyright 02/21/2000, updated 10/10/2001, downloaded from <http://ghp.dwakn.com/contents/csi/>, 2 pages) in view of Gröber et al. (hereinafter Gröber, U.S. Patent No. 6,988,240 filed 08/09/2002, issued 01/17/2006).

Note: The following commentary is provided to further elaborate on the Lennox reference, as it is rather terse. However, the Examiner believes that it either directly or implicitly discloses all of the limitations of the claims, as amended.

The Lennox reference discloses the concept and implementation of "client-side includes". Client-side includes are used in situations where "server-side includes" are not supported. Server-side includes (see example, Pg. 1) when encountered in, for example, an HTML file, cause a server (e.g., a web server) to "include" a file referenced by a "custom" tagging embedded into the HTML file by its creator, or perhaps by a dynamic process that built the page on the server. The key is that server-side includes are performed on a server, then

the server sends the file to a requesting client for rendering and display, for example, on a typical web browser.

Client-side includes, on the other hand, execute on the client; the server (web) simply sends the file "as is" to the client. The client-side include is implemented as an external JavaScript file contained on the client. When a web browser loads the HTML file containing the embedded special tag (<script>), the browser (more specifically a parser component within the browser) interprets the <script> tag, loads and executes the JavaScript code. In this case, the JavaScript code contains a series of "write" commands which cause the HTML coding and content of a "footer" to be embedded into the location where the <script> tag resided. The resulting HTML page is then rendered to the client for display. The resulting displayed HTML file contains its original content plus a customized footer (see Pages 1 and 2 and annotated side note indicators).

In regard to independent Claim 1, Lennox discloses:

- *A method for operating on a client computer for loading a markup language document (Pg. 1 → describes a solution to implement client-side includes using JavaScript), said method comprising the steps of:*
 - *receiving said document comprising a plurality of tags, at least one of said tags being a custom tag (Pgs. 1-2 → web pages (e.g., a web site), containing HTML markup (which contains, for example, a footer, a header, etc. that the page creator wishes to be able to change from time-to-time and an external JavaScript file. The HTML page contains, in the case of a*

footer, the line `<script scr="footer.js"></script>` at the end of each of the HTML pages (see Pg. 1 margin annotation 1). The external JavaScript file present on the client, footer.js, contains a series of lines which, when executed, add HTML lines to the web page containing the footer information (see Pg. 1, margin annotation 2).

Note: it is assumed that browsers typically available at the time of invention, and of the Lennox reference (e.g., Internet Explorer, Netscape) were designed to parse and interpret the various HTML tags in a known fashion (e.g., recognize and properly interpret a `<script>` tag).

The Lennox reference therefore, would have been expected to cause a browser's parser to process its `<script>` tags in a manner consistent with available browsers.

Thus, a typical browser, upon loading an HTML file containing `<script>` tags would have been expected to perform,

- o *parsing said document to determine if certain of said plurality is said custom tag* (as the client browser loads the page, the parser encounters the `<script>` tag (interpreted as a custom tag), recognizes by conventional means (in which script tags are typically interpreted by browser parsers), the `<script>` tag, and loads the file footer.js, which is resident on the client (note, the fact that the path, or lack thereof, accompanying the `scr="footer.js"` attribute would suggest that the file exists on the client).

Lennox fails to disclose:

- *wherein said parsing includes scanning a document object model (DOM) representation of said document to include said executable instructions in place of said custom tag;*

However, Gröber discloses *wherein said parsing includes scanning a document object model (DOM) representation of said document to include said executable instructions in place of said custom tag* (Col. 1, lines 57-65 → Gröber generally discloses a method of enhancing the presentation of a web page on a browser. HTML elements selected to be enhanced are encoded with “tags” or other unique identification. To process the “tags” a player is activated); (Col. 6, lines 4-20; Col. 7, lines 21-43; Fig. 1 → a web browser, per convention, parses the web page and generates a Document Object Model (DOM). The DOM is then accessed by the player, locates enhancement “tags” in the DOM, and performs steps to enhance the presentation of the tagged content. The enhancement steps involve either the generation of additional HTML, or via direct modification of the DOM. If the “tag” for the identified element contains a text, graphic or other information for use in enhancing the presentation (i.e., embedded in the HTML page), the player uses that information accordingly. This can be by direct modification of the DOM though, preferably, it is by generation of further HTML for display by the browser in the conventional manner. If the tag includes an I.D. (unique or otherwise) of the HTML element (or of associated enhancement text, graphics, scripts, etc.), the player looks to an auxiliary store for the enhancement information. In the illustrated embodiment, this is by way of a request to XML or

other database maintained on server or by another digital data processor.

Referring to step G of Fig. 1, the request by the browser to the server includes a UI identifier and/or HTML page identifier. The server uses this to fetch or obtain text, graphics, scripts or other presentation enhancement information from the database. The enhancement is communicated back to the player 20 (or, alternatively, directly to the browser 16). See, step H. The player 20, in turn, generates HTML reflecting the enhancement or modifies the DOM 22 accordingly. See, step I. The web page presentation is thereby enhanced. See, step J. In the event that the enhancements downloaded from the server, e.g., in step H, include "scripts" as noted above, the player 20 executes those in order to provide further web page enhancement that ends, e.g., at script termination or upon user request).

Thus, Gröber discloses *wherein said parsing includes scanning a document object model (DOM) representation of said document to include said executable instructions in place of said custom tag* as scripts are downloaded in place of tags previously added to a web page in order to provide additional functionality to that web page by, for example, adding additional HTML or other content. To do this, Gröber uses a player that scans and modifies a DOM created initially by a web browser.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the disclosures of Lennox and Gröber as both inventions are related to the inclusion of additional content by way of special or custom tags

added to an existing web document. Adding the disclosure of Gröber provides the benefit of scanning and altering the DOM representation of the page by locating "tags" and replacing those tags to include the additional content into the previously generated DOM.

Lennox continues by disclosing:

- *inserting executable instructions into said document at a location of said custom tag, if said custom tag is present (see Page 1, sidenote 2 → the footer.js file executes. The JavaScript statements in this file simply print the footer HTML content into the web page at the location of the <script> tag).*

Lennox fails to disclose:

- *wherein said inserting includes modifying said DOM representation of said document to include said executable instructions in place of said custom tag;*

However, Gröber discloses *wherein said inserting includes modifying said DOM representation of said document to include said executable instructions in place of said custom tag* (Col. 6, lines 4-20; Col. 7, lines 21-43; Fig. 1 → a web browser, per convention, parses the web page and generates a Document Object Model (DOM). The DOM is then accessed by the player, locates enhancement "tags" in the DOM, and performs steps to enhance the presentation of the tagged content. The enhancement steps involve either the

generation of additional HTML, or via direct modification of the DOM. If the "tag" for the identified element contains a text, graphic or other information for use in enhancing the presentation (i.e., embedded in the HTML page), the player uses that information accordingly. This can be by direct modification of the DOM though, preferably, it is by generation of further HTML for display by the browser in the conventional manner. If the tag includes an I.D. (unique or otherwise) of the HTML element (or of associated enhancement text, graphics, scripts, etc.), the player looks to an auxiliary store for the enhancement information. In the illustrated embodiment, this is by way of a request to XML or other database maintained on server or by another digital data processor. Referring to step G of Fig. 1, the request by the browser to the server includes a UI identifier and/or HTML page identifier. The server uses this to fetch or obtain text, graphics, scripts or other presentation enhancement information from the database. The enhancement is communicated back to the player 20 (or, alternatively, directly to the browser 16). See, step H. The player 20, in turn, generates HTML reflecting the enhancement or modifies the DOM 22 accordingly. See, step 1. The web page presentation is thereby enhanced. See, step J. In the event that the enhancements downloaded from the server, e.g., in step H, include "scripts" as noted above, the player 20 executes those in order to provide further web page enhancement that ends, e.g., at script termination or upon user request).

Thus, Gröber discloses *wherein said inserting includes modifying said DOM representation of said document to include said executable instructions in place*

of said custom tag as scripts are downloaded in place of tags previously added to a web page in order to provide additional functionality to that web page by, for example, adding additional HTML or other content. To do this, Gröber uses a player that scans and modifies a DOM created initially by a web browser.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the disclosures of Lennox and Gröber as both inventions are related to the inclusion of additional content by way of special or custom tags added to an existing web document. Adding the disclosure of Gröber provides the benefit of altering the DOM representation of the page by including the additional content into the previously generated DOM.

Lennox continues by disclosing:

- *executing said instructions; and rendering said document on a display device* (the footer.js file executes (see Pg. 1, side note 2. The JavaScript statements in this file simply print the footer HTML content into the web page at the location of the <script> tag).

In regard to dependent Claim 2, Lennox discloses:

- *said markup language is HTML* (Pgs. 1-2 → portrays tagging that is indicative of what a user would have expected to find in a typical HTML page at the time of invention, see for example, side note 1 on page 1, and paragraph below side note 1).

In regard to dependent Claim 4, Lennox discloses:

- *said document is received over the internet* (Pg. 1 → in discussions of both server and client-side includes is mentioned a server sending a page to a client typically over a network as one would expect of a client-server system).

In regard to dependent Claim 5, Lennox discloses:

- *the step of rendering further comprises using a browser* (Pg. 2 → discusses the use of a browser “getting” the footer.js file and writing its contents to the page).

In regard to independent Claim 6, Lennox discloses:

- *A method for creating and providing a markup language document to a receiving computer* (Pg. 1 → describes a solution to implement client-side includes using JavaScript into a web document resident on a server and requested by a user), *said method comprising the steps of:*
 - *loading said document into a memory on a server computer, said document containing conventional tags and at least one custom tag, said custom tag associated with machine-executable instructions resident on a receiving computer in communication with said network; providing said document from said server computer to said receiving computer over said network; said receiving computer identifying said custom tag ...* (Pg. 1, 2nd Paragraph → in comparing Server Side Includes (SSI) with Client Side Includes (CSI), both are described as being contained in pages served by

a server to a client, 4th Paragraph → the notion in item 1 (see hand-numbering on Pg. 1 of Lennox) is that of a client-resident JavaScript executable that is executed in a conventional manner on the client upon the receipt of the web page containing the <script> tag and its detection by the browser).

Lennox fails to disclose:

- *wherein said identifying includes scanning a document object model (DOM) representation of said document in said receiving computer for the presence of said custom tag.*

However, Gröber discloses *wherein said identifying includes scanning a document object model (DOM) representation of said document in said receiving computer for the presence of said custom tag* (Col. 1, lines 57-65 → Gröber generally discloses a method of enhancing the presentation of a web page on a browser. HTML elements selected to be enhanced are encoded with "tags" or other unique identification. To process the "tags" a player is activated); (Col. 6, lines 4-20; Col. 7, lines 21-43; Fig. 1 → a web browser, per convention, parses the web page and generates a Document Object Model (DOM). The DOM is then accessed by the player, locates enhancement "tags" in the DOM, and performs steps to enhance the presentation of the tagged content. The enhancement steps involve either the generation of additional HTML, or via direct modification of the DOM. If the "tag" for the identified element contains a text, graphic or other information for use in enhancing the presentation (i.e., embedded in the HTML

page), the player uses that information accordingly. This can be by direct modification of the DOM though, preferably, it is by generation of further HTML for display by the browser in the conventional manner. If the tag includes an I.D. (unique or otherwise) of the HTML element (or of associated enhancement text, graphics, scripts, etc.), the player looks to an auxiliary store for the enhancement information. In the illustrated embodiment, this is by way of a request to XML or other database maintained on server or by another digital data processor. Referring to step G of Fig. 1, the request by the browser to the server includes a UI identifier and/or HTML page identifier. The server uses this to fetch or obtain text, graphics, scripts or other presentation enhancement information from the database. The enhancement is communicated back to the player 20 (or, alternatively, directly to the browser 16). See, step H. The player 20, in turn, generates HTML reflecting the enhancement or modifies the DOM 22 accordingly. See, step 1. The web page presentation is thereby enhanced. See, step J. In the event that the enhancements downloaded from the server, e.g., in step H, include "scripts" as noted above, the player 20 executes those in order to provide further web page enhancement that ends, e.g., at script termination or upon user request).

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the disclosures of Lennox and Gröber as both inventions are related to the inclusion of additional content by way of special or custom tags added to an existing web document. Adding the disclosure of Gröber provides

the benefit of scanning and altering the DOM representation of the page by locating “tags” and replacing those tags to include the additional content into the previously generated DOM.

Lennox continues by disclosing:

- *inserting said machine-executable instructions into said document at a location associated with said custom tag (see Page 1, side note 2 → the footer.js file executes. The JavaScript statements in this file simply print the footer HTML content into the web page at the location of the <script> tag).*

Lennox fails to disclose:

- *wherein said inserting includes modifying said DOM representation of said document to include said machine-executable instructions in place of said custom tag.*

However, Gröber discloses *wherein said inserting includes modifying said DOM representation of said document to include said executable instructions in place of said custom tag* (Col. 6, lines 4-20; Col. 7, lines 21-43; Fig. 1 → a web browser, per convention, parses the web page and generates a Document Object Model (DOM). The DOM is then accessed by the player, locates enhancement “tags” in the DOM, and performs steps to enhance the presentation of the tagged content. The enhancement steps involve either the generation of additional HTML, or via direct modification of the DOM. If the “tag” for the identified element contains a text, graphic or other information for

use in enhancing the presentation (i.e., embedded in the HTML page), the player uses that information accordingly. This can be by direct modification of the DOM though, preferably, it is by generation of further HTML for display by the browser in the conventional manner. If the tag includes an I.D. (unique or otherwise) of the HTML element (or of associated enhancement text, graphics, scripts, etc.), the player looks to an auxiliary store for the enhancement information. In the illustrated embodiment, this is by way of a request to XML or other database maintained on server or by another digital data processor. Referring to step G of Fig. 1, the request by the browser to the server includes a UI identifier and/or HTML page identifier. The server uses this to fetch or obtain text, graphics, scripts or other presentation enhancement information from the database. The enhancement is communicated back to the player 20 (or, alternatively, directly to the browser 16). See, step H. The player 20, in turn, generates HTML reflecting the enhancement or modifies the DOM 22 accordingly. See, step 1. The web page presentation is thereby enhanced. See, step J. In the event that the enhancements downloaded from the server, e.g., in step H, include "scripts" as noted above, the player 20 executes those in order to provide further web page enhancement that ends, e.g., at script termination or upon user request).

Thus, Gröber discloses *wherein said inserting includes modifying said DOM representation of said document to include said executable instructions*

in place of said custom tag as scripts are downloaded in place of tags previously added to a web page in order to provide additional functionality to that web page by, for example, adding additional HTML or other content. To do this, Gröber uses a player that scans and modifies a DOM created initially by a web browser.

It would have been obvious to one of ordinary skill in the art at the time of invention to combine the disclosures of Lennox and Gröber as both inventions are related to the inclusion of additional content by way of special or custom tags added to an existing web document. Adding the disclosure of Gröber provides the benefit of altering the DOM representation of the page by including the additional content into the previously generated DOM.

- o *executing said instructions; and rendering said document on a display device* (the footer.js file executes (see Pg. 1, side note 2. The JavaScript statements in this file simply print the footer HTML content into the web page at the location of the <script> tag).

In regard to dependent Claim 7, Lennox discloses:

- *said markup language is HTML* (Pgs. 1-2;→ portrays tagging that is indicative of what a user would have expected to find in a typical HTML page at the time of invention, see for example, side note 1 on page 1, and paragraph below side note 1).

In regard to dependent Claim 8, Lennox discloses:

- *said receiving computer is a client computer (Pgs. 1-2; → client-side include is implemented as a JavaScript file located, and executed on a client machine).*

In regard to dependent Claim 10, Lennox discloses:

- *said network is an Internet protocol (IP) network (Pgs. 1-2; → a client-server system implies a network, web pages are being served from servers to clients, the pages contain <script> tags which are understood by browsers on clients. Also, typical client-server systems of the type inferred by Lennox were IP networks).*

In regard to dependent Claim 11, Lennox discloses:

- *said receiving computer renders said document on a display device using a browser (Pgs. 1-2; → browser on client loads HTML page from server containing <script> tag, loads and executes JavaScript which prints additional HTML code for a footer for the page which the browser then renders and displays on the client).*

In regard to Claim 12, Claim 12 merely recites an apparatus for performing the method of claim 1. Thus, Lennox discloses every limitation of Claim 12, as indicated in the above rejection for Claim 1.

In regard to Claims 17 and 18, Claims 17 and 18 merely recite a computer program product for performing the method of claims 1 and 2, respectively. Thus, Lennox discloses every limitation of Claims 17 and 18, as indicated in the above rejection for Claims 1 and 2.

In regard to dependent Claim 19, Lennox discloses:

- *wherein said display device is associated with a client device* (Pg. 1 → including an external JavaScript file in our (web) pages. But, we need some way of writing the HTML and not the JavaScript to the page).

In regard to dependent Claim 20, Lennox discloses:

- *said network is an Internet protocol (IP) network* (Pgs. 1-2 → a client-server system implies a network, web pages are being served from servers to clients, the pages contain <script> tags which are understood by browsers on clients. Also, typical client-server systems of the type inferred by Lennox were IP networks).

Response to Arguments

9. Applicants argue that the prior art of Lennox fails to disclose or suggest any method or system for *loading a markup language document*, including: *receiving said document comprising a plurality of tags, at least one of said tags being a custom tag; parsing said document to determine if certain of said plurality is said custom tag,*

Art Unit: 2176

wherein said parsing includes scanning a document object model (DOM) representation of said document for the presence of said custom tag; inserting executable instructions into said document at a location of said custom tag, if said custom tag is present, wherein said inserting includes modifying said DOM representation of said document to include said executable instructions in place of said custom tag; executing said instructions; and rendering said document on a display device. (emphasis added) as in the present independent claim 1.

10. Lennox discloses nothing about processing a DOM representation of a document in order to identify and/or replace a detected custom tag within the DOM with machine-executable instructions, as in the present independent claims.

11. The Examiner agrees to the extent that Lennox doesn't explicitly mention operations with respect to a DOM. It is noted however that a typical browser at the time of invention parsed and generated a DOM of a requested web page prior to rendering it.

12. The Examiner presents a new rejection based on the prior art of Lennox in view of Gröber. Gröber makes explicit use of a DOM and modifies it according to a custom tag being identified in a received web document.

Conclusion

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP

Art Unit: 2176

§ 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

14. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to James H. Blackwell whose telephone number is 571-272-4089. The examiner can normally be reached on 8-5 M-F.

16. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on 571-272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2176

17. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

James H. Blackwell
11/26/2007

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100